ASSIGNMENT-4

Code:

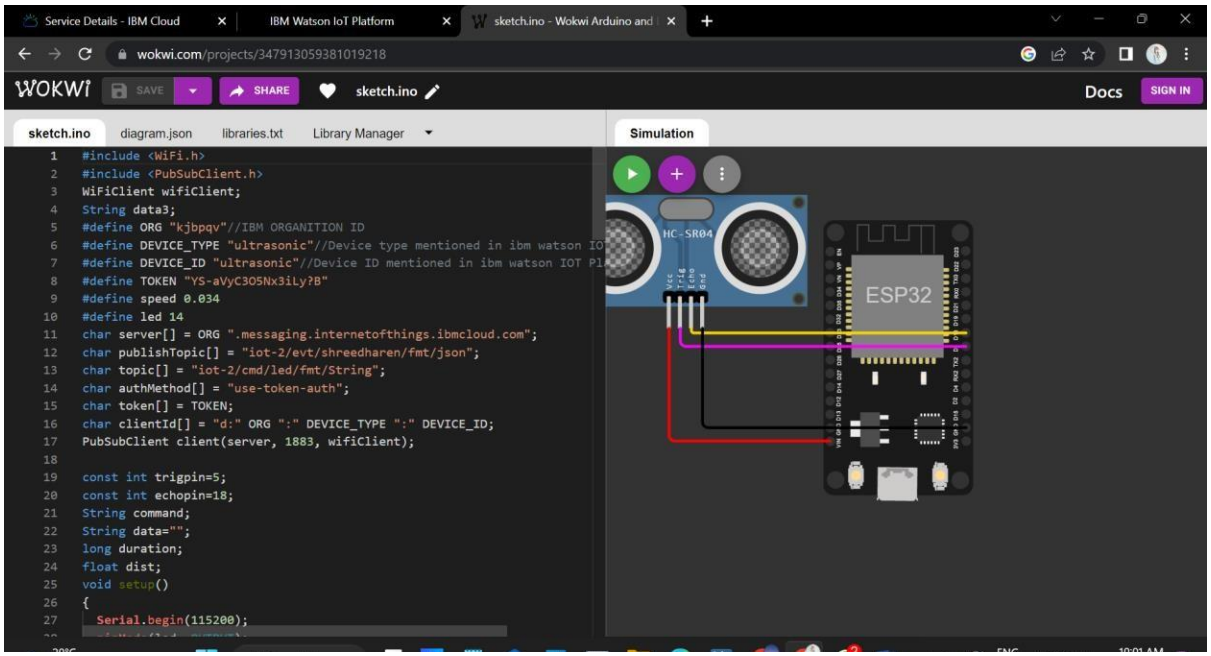
```
#include  <WiFi.h>
#include
<PubSubClient.h>
WiFiClient  wifiClient;
String data3;
#define ORG "kjbpqv"//IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonic"//Device type mentioned in ibm watson IOT
Platform#define DEVICE_ID "ultrasonic"//Device ID mentioned in ibm watson IOT
Platform #define TOKEN "YS-aVyC3O5Nx3iLy?B"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/shreedharen/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);



const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
float dist;
```

```
void setup()
{
 Serial.begin(115200);
 pinMode(led, OUTPUT);
 pinMode(trigpin,
 OUTPUT);
 pinMode(echopin, INPUT);
 wifiConnect();
 mqttConnect();
}

void loop() {
 bool isNearby = dist < 100;
 digitalWrite(led, isNearby);

 publishData();
 delay(500);

 if (!client.loop())
  {mqttConnect();
 }
}

void wifiConnect() {
 Serial.print("Connecting to ");
 Serial.print("Wifi");WiFi.begin("Wokwi-
 GUEST", "", 6);
 while (WiFi.status() !=
  WL_CONNECTED) {delay(500);
  Serial.print(".");
 }
 Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}
```

```
void mqttConnect() {
 if (!client.connected()) {
  Serial.print("Reconnecting MQTT client to ");
  Serial.println(server);while (!client.connect(clientId, authMethod,
  token)) { Serial.print(".");
   delay(500);
  }
  initManagedDevice();
  Serial.println();
 }
}

void initManagedDevice() {
 if (client.subscribe(topic))
 {
  // Serial.println(client.subscribe(topic));
  Serial.println("IBM subscribe to cmd OK");
 } else {
  Serial.println("subscribe to cmd FAILED");
 }
}
void publishData()
{
 digitalWrite(trigpin,LOW);
 digitalWrite(trigpin,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigpin,LOW);
 duration=pulseIn(echopin,HIGH)
 ;dist=duration*speed/2;
 if(dist<100){
  String payload = "{\"Alert Distance\":";
```

```
 payload +=
 dist;payload +=
 "}";

 Serial.print("\n");
 Serial.print("Sending payload:
 ");Serial.println(payload);
 if (client.publish(publishTopic, (char*) payload.c_str()))
  {Serial.println("Publish OK");
 }

}
 if(dist>100){
 String payload =
 "{\"Distance\":";payload += dist;
 payload += "}";

 Serial.print("\n");
 Serial.print("Sending payload:
 ");Serial.println(payload);
  if(client.publish(publishTopic, (char*) payload.c_str()))
  {Serial.println("Publish OK");
 }else {
  Serial.println("Publish FAILED");
 }

}



}
```

Output:


Service Details - IBM Cloud
IBM Watson IoT Platform
sketch.ino - Wokwi Arduino and
wokwi.com/projects/347913059381019218
WOKWI
SAVE
SHARE
sketch.ino
Docs
SIGN IN
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "kjbpqv"//IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonic"//Device type mentioned in ibm watson IO
#define DEVICE_ID "ultrasonic"//Device ID mentioned in ibm watson IOT Pl
#define TOKEN "YS-aVyC3O5Nx3iLy?B"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/shreedharen/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
float dist;
void setup()
{
Serial.begin(115200);
HC-SR04
ESP32


WOKWI
SAVE
SHARE
sketch.ino
Docs
Simulation
00:50.199
99%
ifi....WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to xhlz7n.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK
Sending payload: {"Distance":399.92}
Publish OK
Sending payload: {"Distance":399.92}
Publish OK
Sending payload: {"Distance":399.96}
Publish OK
Sending payload: {"Distance":399.96}
Publish OK
Sending payload: {"Distance":399.94}
Publish OK